# Das2json Overview

Paul Tarvydas, May 14, 2024

# Das2json Compiler

swib.ohm

```
swib {
  main = spaces rule rule*
  rule = "<" spaces rulename "=" spaces pattern+
  pattern = stringMatch | endop | cond | peekcond | cycle | rulecall

  endop = "_end" spaces

  rulename = name spaces

  rulecall = name spaces

  stringMatch = string spaces

  string = dq notdq* dq
  dq = "\""
  notdq =
    | "\\" any -- escaped
    | ~dq any -- raw

  cond = "[" spaces condClause+ "]" spaces
  peekcond = "[+" spaces peekCondClause+ "]" spaces
  peekCondClause = "|" spaces peekCondMatch ":" spaces action*
  condClause = "|" spaces condMatch ":" spaces action*
  condMatch =
    | string -- string
    | endop -- endop
    | "*" -- else
  peekCondMatch =
    | string -- string
    | endop -- endop
    | "*" -- else
  action = break | acceptAndAppend | pattern
  break = "_break" spaces
  acceptAndAppend = "." spaces

  cycle = "<<<" spaces pattern+ ">>>" spaces

  name = firstLetter moreLetter*
  firstLetter = letter | "_"
  moreLetter = digit | firstLetter
}
```

swib.rwr

```
swib {
  main [spaces firstRule moreRule*] = '+firstRule+moreRule+

import receptor
_r = receptor.Receptor ()
[+firstRule+] (_r)
s = _r.pop_return_value ()
print (s)

  rule [_colon spaces rulename _eq spaces2 pattern+] ='
def +rulename+ (_r):(-
    _r.push_new_string ()
    _r.begin_breadcrumb ("+rulename+")
    +pattern+_r.end_breadcrumb ("+rulename+")
    return _r.return_string_pop ()-)\n'

  pattern [p] = '+p+'

  endop [_end spaces] = '_r.eof ()\n'

  rulename [name spaces] = '+name+'

  rulecall [name spaces] = '+name+ (_r)\n_r.append_returned_string ()\n'

  stringMatch [s ws] = '_r.need_and_append (+s+)\n'

  string [dq1 notdq* dq2] = '"+notdq+"'
  dq [c] = '\"'
  notdq_escaped [_bs c] = '+_bs++c+'
  notdq_raw [c] = '+c+'

  cond [lb spaces1 condClause+ rb spaces2] = 'if False:(-\npass-)\n+condClause+'
  condClause [_bar spaces1 condMatch _colon spaces2 action+] = 'elif +condMatch+:(-\n+action+-)'
  condMatch_string [s] = '_r.maybe_append (+s+)'
  condMatch_endop [x] = '_r.eof ()'
  condMatch_else [x] = 'True'

  peekcond [lb spaces1 condClause+ rb spaces2] = 'if False:(-\npass-)\n+condClause+'
  peekCondClause [_bar spaces1 condMatch _colon spaces2 action+] = 'elif +condMatch+:(-\n+action+-)'
  peekCondMatch_string [s] = '_r.peek (+s+)'
  peekCondMatch_endop [x] = '_r.eof ()'
  peekCondMatch_else [x] = 'True'

  action [a] = '+a+'
  break [_break spaces] = 'break\n'
  acceptAndAppend [_dot spaces] = '_r.accept_and_append ()\n'

  cycle [lb spaces1 pattern+ rb spaces2] = 'while True:(-\n+pattern+-)\n'

  name [firstLetter moreLetter*] = '+firstLetter+moreLetter+'
  firstLetter [c] = '+c+'
  moreLetter [c] = '+c+'

}
```

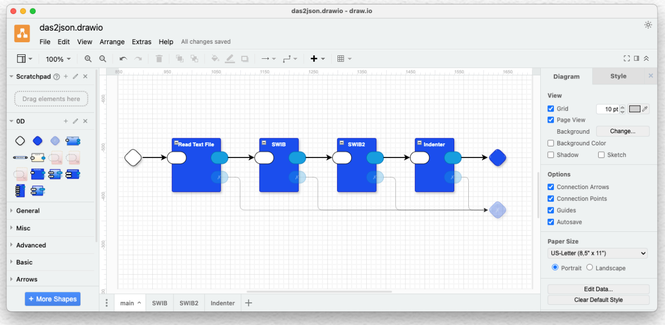
defname.ohm

```
defname {
  main = text+
  text =
    | defName -- match
    | any -- other
  defName = "/" spaces "def" spaces name spaces through<closeindent> spaces "/"
  through<s> = (~s any)+  s
  name = letter alnum* ~alnum
  closeindent = "-)"
}
```

defname.rwr

```
defname {
  main [x+] = '«x»'
  text_match [x] = '«x»'
  text_other [x] = '«x»'
  defName  [lb spaces1 _def spaces2 name spaces3 misc spaces4 rb] = '«name»'
  through [misc s] = ''
  name [letter alnum*] = '«letter»«alnum»'
  closeindent [rb] = ''
}
```

das2json.drawio - draw.io




## das2json.swib

das2json.swib

```
: Das2json =
  XML Spaces _end

: XML =
  Spaces "<" Stuff
  [
    | ">": Content "</" Stuff ">"
    | "/>":
  ]

: Content =
  <<<
    Spaces
    [*
      | "</": _break
      | "<": XML
      | *: Stuff
    ]
  >>>

: Attributes =
  <<<
    [*
      | ">": _break
      | "/>": _break
      | _end: _break
      | *: Stuff
    ]
  >>>

: Stuff =
  <<<
    [*
      | ">": _break
      | "<": _break
      | "/>": _break
      | _end: _break
      | *: .
    ]
  >>>

: Spaces =
  <<<
    [*
      | " ": .
      | "\t": .
      | "\n": .
      | *: _break
    ]
  >>>

: String =
  "\"" NotDquotes "\""

: NotDquotes =
-:---  das2json.swib   Top   L49   Git:main   (Fundamental)
```

0D

## das2json.py

regression.py

```
def Das2json (_r):
    _r.push_new_string ()
    _r.begin_breadcrumb ("Das2json")
    XML (_r)
    _r.append_returned_string ()
    Spaces (_r)
    _r.append_returned_string ()
    _r.eof ()
    _r.end_breadcrumb ("Das2json")
    return _r.return_string_pop ()

def XML (_r):
    _r.push_new_string ()
    _r.begin_breadcrumb ("XML")
    Spaces (_r)
    _r.append_returned_string ()
    _r.need_and_append ("<")
    Stuff (_r)
    _r.append_returned_string ()
    if False:
        pass
    elif _r.maybe_append (">"):
        Content (_r)
        _r.append_returned_string ()
        _r.need_and_append ("</")
        Stuff (_r)
        _r.append_returned_string ()
        _r.need_and_append (">")
        pass
    elif _r.maybe_append ("/>"):
        pass
    _r.end_breadcrumb ("XML")
    return _r.return_string_pop ()

def Content (_r):
    _r.push_new_string ()
    _r.begin_breadcrumb ("Content")
    while True:
        Spaces (_r)
        _r.append_returned_string ()
        if False:
            pass
        elif _r.peek ("</"):
            break
            pass
        elif _r.peek ("<"):
            XML (_r)
            _r.append_returned_string ()
            pass
        elif True:
-:---  regression.py   Top   L37   Git:main   (Python E
```